

Doxxygen_MPDASTARONLINE_DB Reference Manual

Generated by Doxygen 1.3.7

Wed Sep 6 11:30:57 2006

Contents

1	Doxygen_MPDAUTHONLINE_DB Namespace Index	1
1.1	Doxygen_MPDAUTHONLINE_DB Namespace List	1
2	Doxygen_MPDAUTHONLINE_DB Class Index	3
2.1	Doxygen_MPDAUTHONLINE_DB Class List	3
3	Doxygen_MPDAUTHONLINE_DB File Index	5
3.1	Doxygen_MPDAUTHONLINE_DB File List	5
4	Doxygen_MPDAUTHONLINE_DB Namespace Documentation	7
4.1	std Namespace Reference	7
5	Doxygen_MPDAUTHONLINE_DB Class Documentation	9
5.1	CndDbSender Class Reference	9
6	Doxygen_MPDAUTHONLINE_DB File Documentation	21
6.1	CndDbSender.cc File Reference	21
6.2	CndDbSender.hh File Reference	23
6.3	senderMain.cc File Reference	24

Chapter 1

Doxxygen_MPDASTARONLINE_DB Namespace Index

1.1 Doxygen_MPDASTARONLINE_DB Namespace List

Here is a list of all namespaces with brief descriptions:

std	7
-------------------------------	-------------------

Chapter 2

Doxxygen_MPDASTARONLINE_DB Class Index

2.1 Doxygen_MPDASTARONLINE_DB Class List

Here are the classes, structs, unions and interfaces with brief descriptions:

CndDbSender	9
-------------	-------	---

Chapter 3

Doxxygen_MPDASTARONLINE_DB File Index

3.1 Doxygen_MPDASTARONLINE_DB File List

Here is a list of all files with brief descriptions:

CndDbSender.cc	21
CndDbSender.hh	23
senderMain.cc	24

Chapter 4

Doxxygen_MPDU_{STAR}_ONLINE_DB Namespace Documentation

4.1 std Namespace Reference

Chapter 5

Doxxygen_MP_D_STAR_ONLINE_DB Class Documentation

5.1 CndDbSender Class Reference

```
#include <CndDbSender.hh>
```

Public Member Functions

- `CndDbSender()`
- virtual `~CndDbSender()`
- virtual void `setBaseName` (const char *baseName)
- virtual void `setDomainName` (const char *domain)
- virtual void `appendSubDir` (const char *domain)
- virtual const char * `printBaseName` ()
- virtual const char * `printDomainName` ()
- virtual const char * `printBaseDir` ()
- virtual void `cd` (const char *localDir)
- virtual bool `init` (const char *baseName)
- virtual bool `loadControls` (const char *name, const char *value)
- virtual bool `loadUserControls` (const char *name, const char *value)=0
- virtual unsigned int `sleepTime` ()
- virtual bool `wireRequired` ()
- virtual bool `hasBackLog` ()
- virtual void `addBackLog` (unsigned int timestamp, int opt=0)
- virtual bool `cleanBackLog` ()
- virtual bool `updateDb` ()
- virtual void `initQuery` ()=0
- virtual bool `queryData` ()=0
- virtual bool `readData` (const char *fileName)=0
- virtual bool `updateDb` (const char *fileName)=0
- virtual bool `sendMess` (const char *m1, const char *m2, StDbMessLevel level, int line, const char *cname, const char *mname)
- virtual bool `sendMess` (const char *m, StDbMessLevel level, int line, const char *cname, const char *mname)

Protected Member Functions

- void [cycleLogs \(\)](#)
- bool [zeroOutBackLog \(\)](#)
- void [setEmailTo \(const char *addresses\)](#)
- char * [now \(\)](#)

Protected Attributes

- StDbManager * [mgr](#)
- StDbConfigNode * [node](#)
- StDbMessService * [messenger](#)
- char * [mbaseDir](#)
- char * [mdomain](#)
- char * [mibName](#)
- char * [emailList](#)
- char * [configFile](#)
- char * [logFile](#)
- char * [dataFile](#)
- char * [queryFile](#)
- ofstream [mlogs](#)
- unsigned int [sleepTime](#)
- unsigned int [forceWrite](#)
- int [debugFlag](#)
- unsigned int [lastFullWrite](#)
- unsigned int [writeTime](#)
- unsigned int * [backLogTimes](#)
- unsigned int [numBackLogged](#)
- char [cTime](#) [64]

5.1.1 Constructor & Destructor Documentation

5.1.1.1 CndDbSender::CndDbSender ()

constructor & destructor

Definition at line 19 of file CndDbSender.cc.

```

19 : node(0), messenger(0), mibName(0), emailList(0), sleepTime(500), forceWrite
20
21     mbaseDir=new char[strlen("/online/production/database/Run_5/dbSenders/")+1];
22 //MPD's change
23     ostringstream mbb;
24     mbb<<"/online/production/database/Run_5/dbSenders/";
25 //    mbaseDir = (mbb.str()).c_str();
26     strcpy(mbaseDir,(mbb.str()).c_str());
27     // strcpy(mbaseDir,"/online/production/database/Run_5/dbSenders/");
28
29 }
```

5.1.1.2 CndDbSender::~CndDbSender () [virtual]

Definition at line 31 of file CndDbSender.cc.

```

31      {
32      if(node) delete node;
33      delete [] mbaseDir;
34      delete [] mdomain;
35      if(mbaseName) delete [] mbaseName;
36      delete [] configFile;
37      delete [] logFile;
38      delete [] dataFile;
39      delete [] queryFile;
40      if(backLogTimes) delete [] backLogTimes;
41  }
```

5.1.2 Member Function Documentation

5.1.2.1 void CndDbSender::addBackLog (unsigned int *timestamp*, int *opt* = 0) [virtual]

Definition at line 181 of file CndDbSender.cc.

```

181                                     {
182 #define __METHOD__ "addBackLog(timestamp)"
183
184     sendMess("Adding backlog",dbMWarn,__LINE__,__CLASS__,__METHOD__);
185     unsigned int* tmp = new unsigned int[numBackLogged+1];
186     if(numBackLogged){
187         memcpy(tmp,backLogTimes,numBackLogged*sizeof(unsigned int));
188         delete [] backLogTimes;
189     }
190     tmp[numBackLogged]=timestamp;
191     numBackLogged++;
192     backLogTimes=tmp;
193
194     if(!opt){ // except for startup, copy file over
195         //char systemCmd[1024];
196         ostringstream scmd;
197         scmd<< " cp "<<dataFile<<" "<<dataFile<<". "<<timestamp;
198
199     if(system((scmd.str()).c_str())){
200         sendMess("command failed =", (scmd.str()).c_str(),dbMErr,__LINE__,__CLASS__,__METHOD__);
201         numBackLogged--;
202     }
203 }
204
205 #undef __METHOD__
206 }
```

5.1.2.2 void CndDbSender::appendSubDir (const char * *domain*) [virtual]

Definition at line 67 of file CndDbSender.cc.

```

67
68     ostringstream tb;
69     tb<<mbaseDir<<domain<<"/";
70     delete [] mbaseDir;
71 //MPD changes
72     mbaseDir = new char [strlen((tb.str()).c_str()) + 1];
73     strcpy(mbaseDir,(tb.str()).c_str());
74 }
```

5.1.2.3 void CndDbSender::cd (const char * *localDir*) [virtual]

Definition at line 76 of file CndDbSender.cc.

```

76
77     delete [] mbaseDir;
78     ostringstream bd;
79     bd<<localDir<<"/";
80 //MPD changes
81     mbaseDir = new char [strlen((bd.str()).c_str()) + 1];
82     strcpy(mbaseDir,(bd.str()).c_str());
83 }
```

5.1.2.4 bool CndDbSender::cleanBackLog () [virtual]

Definition at line 210 of file CndDbSender.cc.

```

210
211 #define __METHOD__ "cleanBackLog()"
212
213     int j=0;
214     for(int i=0; i<(int)numBackLogged; i++){
215         //char backLogFile[512];
216         ostringstream fn;
217         fn<<dataFile<<"."<<backLogTimes[i];
218         writeTime=backLogTimes[i];
219         if(!updateDb((fn.str()).c_str())) break;
220         //char rmCmd[512];
221         ostringstream rmf;
222         rmf<<"rm -f "<<fn.str();
223         system((rmf.str()).c_str());
224         j++;
225     }
226
227     int left = numBackLogged-j;
228     if(!left) return zeroOutBackLog();
229
230     // reset counters if we got some of them
231     if(j) {
232         unsigned int* tmp = new unsigned int[left];
233         for(int k=0; k<left; k++)tmp[k]=backLogTimes[j+k];
234         delete [] backLogTimes;
235         backLogTimes=tmp;
236         numBackLogged=left;
237     }
238
239     // if we get here, log a warning message
240     //char message[256];
241     ostringstream ms;
242     ms<<"Only cleaned up "<<j<<" out of "<<numBackLogged<<" data files ";
243
244     return sendMess((ms.str()).c_str(),dbMWarn,__LINE__,__CLASS__,__METHOD__);
245
246 #undef __METHOD__
247 }
```

5.1.2.5 void CndDbSender::cycleLogs () [protected]

Definition at line 316 of file CndDbSender.cc.

```

316
317
318 DIR* mydir = opendir(mbaseDir);
319 dirent *p=0;
320 bool done=false;
321 char names[5][64];
322 int j=0;
323 char* dot;
324
325 // find the set of log files
326 while(!done){
327     p=readdir(mydir);
328     if(!p){
329         done=true;
330     } else {
331         if( strstr(p->d_name,mbaseName) ){
332             if(strstr(p->d_name,".log.") && !(strstr(p->d_name,"err")) && j<4){
333                 strcpy(names[j],p->d_name);
334                 j++;
335             }
336             if(!(strstr(p->d_name,"log")) && !(strstr(p->d_name,"list")) &&
337                 !(strstr(p->d_name,"conf"))){
338                 if( (dot=strrchr(p->d_name,".")) ){
339                     dot++;
340                     unsigned int timestamp = (unsigned int)atoi(dot);
341                     if(timestamp>0)addBackLog(timestamp,1);
342                 }
343             }
344         }
345     }
346 }
347 }
348 }
349
350 // push the logs up e.g. .log.3->.log.4, .log.2->.log.1,....log->.log.0
351 for(int k=j;k>=0;k--){
352     // char command[256];
353     ostringstream cmd;
354     cmd<<"mv "<<mbaseDir<<mbaseName<<".log";
355     if(k>0)cmd<<"."<<k-1;
356     cmd<<" "<<mbaseDir<<mbaseName<<".log."<<k;
357     system((cmd.str()).c_str());
358 }
359
360
361 }

```

5.1.2.6 bool CndDbSender::hasBackLog () [inline, virtual]

Definition at line 131 of file CndDbSender.hh.

```
131 { return (numBackLogged) ? true : false; }
```

5.1.2.7 bool CndDbSender::init (const char * *baseName*) [virtual]

Definition at line 89 of file CndDbSender.cc.

```

89
90 #define __METHOD__ "init(baseName)"
91
92     setBaseName(baseName);

```

```

93     ostringstream confF;
94     confF<<mbaseDir<<baseName<<".conf";
95     ostringstream logF;
96     logF<<mbaseDir<<baseName<<".log";
97     ostringstream dataF;
98     dataF<<mbaseDir<<baseName;
99     ostringstream queryF;
100    queryF<<mbaseDir<<baseName<<".list";
101
102 //MPD changes
103    configFile = new char [strlen((confF.str()).c_str()) + 1];
104    strcpy(configFile,(confF.str()).c_str());
105
106    logFile = new char [strlen((logF.str()).c_str()) + 1];
107    strcpy(logFile,(logF.str()).c_str());
108
109    dataFile = new char [strlen((dataF.str()).c_str()) + 1];
110    strcpy(dataFile,(dataF.str()).c_str());
111
112    queryFile = new char [strlen((queryF.str()).c_str()) + 1];
113    strcpy(queryFile,(queryF.str()).c_str());
114
115 // set the log file to the messenger
116    cycleLogs();
117    mlogs.open(logFile);
118    messenger = new StDbMessenger(mlogs);
119
120 // add the messenger to the db-api
121    mgr=StDbManager::Instance();
122    mgr->setMessenger(messenger);
123
124 // now read in the init file to set control variables
125    char line[256];
126    char* tmp;
127    char* tmp2;
128
129    ifstream from(configFile);
130    if (!from) return sendMess("opening config file=",configFile,dbMFatal,__LINE__,__CLASS__,__METHOD__);
131
132    bool done = false;
133    while(!done){
134        if(from.eof())
135            done=true;
136        else {
137            from.getline(line,256);
138            if(line[0]=='#' || !(tmp=strstr(line,"="))) continue;
139            *tmp='\0';
140            tmp++;
141            if((tmp2=strstr(tmp,"#"))){ *tmp2='\0'; }
142            loadControls(line,tmp);
143        }
144    }
145    from.close();
146
147 return true;
148 #undef __METHOD__
149 }
```

5.1.2.8 virtual void CndDbSender::initQuery () [pure virtual]

5.1.2.9 bool CndDbSender::loadControls (const char * name, const char * value) [virtual]

Definition at line 152 of file CndDbSender.cc.

```

152
153
154     if(strstr(name,"sleep")){
155         sleeptime=atoi(value);
156         return true;
157     }
158
159     if(strstr(name,"forceWrite")){
160         forceWrite=atoi(value);
161         return true;
162     }
163
164     if(strstr(name,"debug")){
165         debugFlag=atoi(value);
166         if(debugFlag>=2)messenger->setMessLevel(dbMDebug);
167         return true;
168     }
169
170     return loadUserControls(name,value);
171 }
```

5.1.2.10 virtual bool CndDbSender::loadUserControls (const char * name, const char * value)
 [pure virtual]

5.1.2.11 char * CndDbSender::now () [inline, protected]

Definition at line 112 of file CndDbSender.hh.

```

112
113     time_t t=time(NULL);
114     strcpy(cTime,ctime(&t));
115     char* tmp;
116     if( (tmp=strstr(cTime, "\n")))*tmp='\0';
117     return (char*) cTime;
118 }
```

5.1.2.12 const char * CndDbSender::printBaseDir () [inline, virtual]

Definition at line 123 of file CndDbSender.hh.

```
123 { return mbaseDir; }
```

5.1.2.13 const char * CndDbSender::printBaseName () [inline, virtual]

Definition at line 121 of file CndDbSender.hh.

```
121 { return mbaseName; }
```

5.1.2.14 const char * CndDbSender::printDomainName () [inline, virtual]

Definition at line 122 of file CndDbSender.hh.

```
122 { return mdomain; }
```

5.1.2.15 virtual bool CndDbSender::queryData () [pure virtual]

5.1.2.16 virtual bool CndDbSender::readData (const char **fileName*) [pure virtual]

5.1.2.17 bool CndDbSender::sendMess (const char **m*, StDbMessLevel *level*, int *line*, const char **cname*, const char **mname*) [virtual]

Definition at line 272 of file CndDbSender.cc.

```

272
273
274     if(!messenger) return true;
275     StDbMessLevel saveIt;
276     if(debugFlag==1) {
277         saveIt=messenger->getMessLevel();
278         messenger->setMessLevel(dbMDebug);
279     }
280
281     ostringstream dm; dm<<"<<now()<<"> "<<m;
282     messenger->printMessage((dm.str()).c_str(),level,line,cname,mname);
283     //delete [] (dm.str()).c_str();
284
285     if(level == dbMFatal){
286         if(emailList){
287
288             mlogs.close();
289             char errorFile[256];
290             ostringstream ef;
291             ef<<"/tmp/"<<mibName<<.err";
292             //char createErrorFile[256];
293             ostringstream ecmd;
294             ecmd<<" tail "<<logFile<<" > "<<errorFile;
295             //char mailError[512];
296             ostringstream me;
297             me<<" mailx -s \"<<mibName<<"ERROR\" ";
298             me<<emailList<<" < "<<errorFile<<" >/dev/null &";
299
300             // tail the log into an error file
301             system((ecmd.str()).c_str());
302             // mail off the error file
303             system((me.str()).c_str());
304         }
305
306         exit(0);
307     }
308
309     if(debugFlag==1)messenger->setMessLevel(saveIt);
310
311     if(level==dbMErr) return false;
312     return true;
313 }
```

5.1.2.18 bool CndDbSender::sendMess (const char **m1*, const char **m2*, StDbMessLevel *level*, int *line*, const char **cname*, const char **mname*) [virtual]

Definition at line 255 of file CndDbSender.cc.

```

255
256 // char message[4096];
257 ostringstream mtotal;
258 mtotal<<m1<<" "<<m2;
```

```
259     return sendMess((mtotal.str()).c_str(),level,line,cname,mname);
260 }
```

5.1.2.19 void CndDbSender::setBaseName (const char * *baseName*) [virtual]

Definition at line 54 of file CndDbSender.cc.

```
54 {
55     if(mbaseName) delete [] mbaseName;
56     mbaseName = new char[strlen(baseName)+1];
57     strcpy(mbaseName,baseName);
58 }
```

5.1.2.20 void CndDbSender::setDomainName (const char * *domain*) [virtual]

Definition at line 60 of file CndDbSender.cc.

```
60 {
61     if(mdomain) delete [] mdomain;
62     mdomain = new char[strlen(domain)+1];
63     strcpy(mdomain,domain);
64     appendSubDir(mdomain);
65 }
```

5.1.2.21 void CndDbSender::setEmailTo (const char * *addresses*) [protected]

Definition at line 48 of file CndDbSender.cc.

```
48 {
49     emailList = new char[strlen(addresses)+1];
50     strcpy(emailList,addresses);
51 }
```

5.1.2.22 unsigned int CndDbSender::sleepTime () [inline, virtual]

Definition at line 130 of file CndDbSender.hh.

```
130 { return sleeptime; }
```

5.1.2.23 virtual bool CndDbSender::updateDb (const char * *fileName*) [pure virtual]

5.1.2.24 bool CndDbSender::updateDb () [virtual]

Definition at line 174 of file CndDbSender.cc.

```
174 { return updateDb(dataFile); }
```

5.1.2.25 bool CndDbSender::writeRequired () [inline, virtual]

Definition at line 132 of file CndDbSender.hh.

```
132 {  
133     return ((writeTime-lastFullWrite)<forceWrite) ? false : true;  
134 }
```

5.1.2.26 bool CndDbSender::zeroOutBackLog () [inline, protected]

Definition at line 124 of file CndDbSender.hh.

```
124 {  
125     if(backLogTimes) { delete [] backLogTimes; backLogTimes = 0; }  
126     numBackLogged = 0;  
127     return true;  
128 }
```

5.1.3 Member Data Documentation**5.1.3.1 unsigned int* CndDbSender::backLogTimes [protected]**

Definition at line 52 of file CndDbSender.hh.

5.1.3.2 char* CndDbSender::configFile [protected]

Definition at line 37 of file CndDbSender.hh.

5.1.3.3 char CndDbSender::cTime[64] [protected]

Definition at line 59 of file CndDbSender.hh.

5.1.3.4 char* CndDbSender::dataFile [protected]

Definition at line 39 of file CndDbSender.hh.

5.1.3.5 int CndDbSender::debugFlag [protected]

Definition at line 47 of file CndDbSender.hh.

5.1.3.6 char* CndDbSender::emailList [protected]

Definition at line 35 of file CndDbSender.hh.

5.1.3.7 unsigned int CndDbSender::forceWrite [protected]

Definition at line 46 of file CndDbSender.hh.

5.1.3.8 `unsigned int CndDbSender::lastFullWrite` [protected]

Definition at line 49 of file CndDbSender.hh.

5.1.3.9 `char* CndDbSender::logFile` [protected]

Definition at line 38 of file CndDbSender.hh.

5.1.3.10 `char* CndDbSender::mbaseDir` [protected]

Definition at line 31 of file CndDbSender.hh.

5.1.3.11 `char* CndDbSender::mibName` [protected]

Definition at line 33 of file CndDbSender.hh.

5.1.3.12 `char* CndDbSender::mdomain` [protected]

Definition at line 32 of file CndDbSender.hh.

5.1.3.13 `StDbMessService* CndDbSender::messenger` [protected]

Definition at line 28 of file CndDbSender.hh.

5.1.3.14 `StDbManager* CndDbSender::mgr` [protected]

Definition at line 26 of file CndDbSender.hh.

5.1.3.15 `ofstream CndDbSender::mlogs` [protected]

Definition at line 42 of file CndDbSender.hh.

5.1.3.16 `StDbConfigNode* CndDbSender::node` [protected]

Definition at line 27 of file CndDbSender.hh.

5.1.3.17 `unsigned int CndDbSender::numBackLogged` [protected]

Definition at line 53 of file CndDbSender.hh.

5.1.3.18 `char* CndDbSender::queryFile` [protected]

Definition at line 40 of file CndDbSender.hh.

5.1.3.19 unsigned int CndDbSender::sleepTime [protected]

Definition at line 44 of file CndDbSender.hh.

5.1.3.20 unsigned int CndDbSender::writeTime [protected]

Definition at line 50 of file CndDbSender.hh.

The documentation for this class was generated from the following files:

- [CndDbSender.hh](#)
- [CndDbSender.cc](#)

Chapter 6

Doxxygen_MP_D_STAR_ONLINE_DB File Documentation

6.1 CndDbSender.cc File Reference

```
#include "CndDbSender.hh"
#include "StDbManager.hh"
#include "StDbMessenger.hh"
#include <stdlib.h>
#include <dirent.h>
```

Namespaces

- namespace `std`

Defines

- `#define __CLASS__ "CndDbSender"`
- `#define __METHOD__ "init(baseName)"`
- `#define __METHOD__ "addBackLog(timestamp)"`
- `#define __METHOD__ "cleanBackLog()"`

6.1.1 Define Documentation

6.1.1.1 `#define __CLASS__ "CndDbSender"`

Definition at line 11 of file CndDbSender.cc.

- 6.1.1.2 #define __METHOD__ "cleanBackLog()
- 6.1.1.3 #define __METHOD__ "addBackLog(timestamp)"
- 6.1.1.4 #define __METHOD__ "init(baseName)"

6.2 CndDbSender.hh File Reference

```
#include "StDbManager.hh"
#include "StDbConfigNode.hh"
#include "stldb_streams.h"
#include <time.h>
```

Classes

- class [CndDbSender](#)

6.3 senderMain.cc File Reference

```
#include <unistd.h>
#include <sys/stat.h>
#include <iostream.h>
#include <math.h>
```

Functions

- void [runSender](#) (const char *directory)
- int [main](#) (int argc, char *argv[])

6.3.1 Function Documentation

6.3.1.1 int main (int *argc*, char * *argv*[])

Definition at line 17 of file senderMain.cc.

```
17
18
19
20     pid_t pid;
21     if((pid = fork()) < 0) cout<<("Unable to fork process");
22     else if(pid != 0) exit(0);
23     setsid(); //run as session leader
24     chdir("/");
25     umask(0); //change umask
26
27     char* localDirectory = 0;
28
29     for (int i=1; i<argc;i++){
30         if(argv[i][0] == '-'){
31             switch (argv[i][1]) {
32                 case 'c' :
33                     localDirectory = argv[++i];
34                     break;
35                 default :
36                     break;
37             }
38         }
39     }
40     runSender(localDirectory);
41
42
43 return 0;
44 };
```

6.3.1.2 void runSender (const char * *directory*)

Index

~CndDbSender
 CndDbSender, 10
__CLASS__
 CndDbSender.cc, 21
__METHOD__
 CndDbSender.cc, 21, 22

addBackLog
 CndDbSender, 11
appendSubDir
 CndDbSender, 11

backLogTimes
 CndDbSender, 18

cd
 CndDbSender, 11
cleanBackLog
 CndDbSender, 12
CndDbSender, 9
 CndDbSender, 10
CndDbSender
 ~CndDbSender, 10
 addBackLog, 11
 appendSubDir, 11
 backLogTimes, 18
 cd, 11
 cleanBackLog, 12
 CndDbSender, 10
 configFile, 18
 cTime, 18
 cycleLogs, 12
 dataFile, 18
 debugFlag, 18
 emailList, 18
 forceWrite, 18
 hasBackLog, 13
 init, 13
 initQuery, 14
 lastFullWrite, 18
 loadControls, 14
 loadUserControls, 15
 logFile, 19
 mbaseDir, 19
 mbaseName, 19

 mdomain, 19
 messenger, 19
 mgr, 19
 mlogs, 19
 node, 19
 now, 15
 numBackLogged, 19
 printBaseDir, 15
 printBaseName, 15
 printDomainName, 15
 queryData, 15
 queryFile, 19
 readData, 16
 sendMess, 16
 setBaseName, 17
 setDomainName, 17
 setEmailTo, 17
 sleepTime, 17
 sleeptime, 19
 updateDb, 17
 writeRequired, 17
 writeTime, 20
 zeroOutBackLog, 18
CndDbSender.cc, 21
CndDbSender.cc
 __CLASS__, 21
 __METHOD__, 21, 22
CndDbSender.hh, 23
configFile
 CndDbSender, 18
cTime
 CndDbSender, 18
cycleLogs
 CndDbSender, 12

dataFile
 CndDbSender, 18
debugFlag
 CndDbSender, 18

emailList
 CndDbSender, 18

forceWrite
 CndDbSender, 18

hasBackLog
 CndDbSender, 13

init
 CndDbSender, 13

initQuery
 CndDbSender, 14

lastFullWrite
 CndDbSender, 18

loadControls
 CndDbSender, 14

loadUserControls
 CndDbSender, 15

logFile
 CndDbSender, 19

main
 senderMain.cc, 24

mbaseDir
 CndDbSender, 19

mbaseName
 CndDbSender, 19

mdomain
 CndDbSender, 19

messenger
 CndDbSender, 19

mgr
 CndDbSender, 19

mlogs
 CndDbSender, 19

node
 CndDbSender, 19

now
 CndDbSender, 15

numBackLogged
 CndDbSender, 19

printBaseDir
 CndDbSender, 15

printBaseName
 CndDbSender, 15

printDomainName
 CndDbSender, 15

queryData
 CndDbSender, 15

queryFile
 CndDbSender, 19

readData
 CndDbSender, 16

runSender
 senderMain.cc, 24

senderMain.cc, 24

senderMain.cc
 main, 24

runSender, 24

sendMess
 CndDbSender, 16

setBaseName
 CndDbSender, 17

setDomainName
 CndDbSender, 17

setEmailTo
 CndDbSender, 17

sleepTime
 CndDbSender, 17

sleeptime
 CndDbSender, 19

std, 7

updateDb
 CndDbSender, 17

writeRequired
 CndDbSender, 17

writeTime
 CndDbSender, 20

zeroOutBackLog
 CndDbSender, 18